



A Collaborative Tracking Algorithm for Communicating Target in Wireless Multimedia Sensor Networks

Ibtissem Boulanouar, Stéphane Lohier, Abderrezak Rachedi, Gilles Roussel

► To cite this version:

Ibtissem Boulanouar, Stéphane Lohier, Abderrezak Rachedi, Gilles Roussel. A Collaborative Tracking Algorithm for Communicating Target in Wireless Multimedia Sensor Networks. 7th IFIP Wireless and Mobile Networking Conference, Mar 2014, Vilamoura, Algarve, Portugal. Wireless and Mobile Networking, 2014, <10.1109/WMNC.2014.6878883>. <hal-00962257>

HAL Id: hal-00962257

<https://hal-upec-upem.archives-ouvertes.fr/hal-00962257>

Submitted on 26 Aug 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Collaborative Tracking Algorithm for Communicating Target in Wireless Multimedia Sensor Networks

Ibtissem Boulanouar, Stéphane Lohier, Abderrezak Rachedi and Gilles Roussel

Université Paris-Est - Gaspard Monge Computer Science Laboratory (LIGM - UMR 8049)
75420 Champs sur Marne, France
{FirstName. LastName}@univ-mlv.fr

Abstract—In this paper, we address the problem of target tracking in Wireless Multimedia Sensor Networks. Target tracking is usually defined as a two stages process: 1) detecting the presence of the target and 2) locating it. We propose a cluster-based and collaborative tracking algorithm for a signal emitting target with the objective of finding the best trade-off between the energy consumption and the tracking precision. In this algorithm, each cluster component is in charge of specific tasks. More powerful sensors handle the high cost energy tasks and assume inter and intra-cluster collaboration while constraint sensors handle low-cost energy tasks and assume only intra-cluster communication. A probabilistic node selection method is implemented to select the best sensors which participate to the tracking process. A deployment strategy for both sensors is also proposed. Simulation results are presented to evaluate the efficiency of the proposed algorithm. They demonstrate a significant target tracking accuracy improvement and energy consumption reduction comparing to existing algorithms.

Keywords. *Wireless Multimedia Sensor Network, mobile target tracking.*

I. INTRODUCTION

A Wireless Multimedia Sensor Network (WMSN) is a network composed of small, autonomous, low-powered and distributed multimedia sensors able to communicate and exchange multimedia data through wireless links. In such network the sensors are fitted with cameras and microphones in order to sense, capture and process multimedia content. Applications of WMSN are usually habitat and health care monitoring, environmental control, military surveillance and other security applications [1]. While standard Wireless Sensor Networks have omni-directional sensing field, WMSN have directional and orientable one. They can be also very useful in indoor environment, where there is neither standard monitoring systems or communication infrastructures.

Our focus in this work is single target tracking, specially communicating target. Let us specify that communicating targets such as mobile sensors, are equipped with a communication module which allows them to transmit signals and thus, communicate with network infrastructure. The interest for this kind of targets stems from the large number of applications and services that might be achievable. For example, from the

location of the target, automated and personalized services could be offered. Target tracking involves presence detection and localization of the dynamic target throughout the monitoring area. In the context of WMSN, harvested multimedia data such as video frames are used to get more information about the mobile target: identity, shapes, etc. This information can be very useful to provide services. However, this kind of network has some limitations: wireless links are not reliable and data processing and transmission are greedy processes in term of energy. To overcome the energy constrain, only the multimedia sensors in target pathway should be activated.

To relieve the aforementioned limitations, we propose a distributed and collaborative target tracking algorithm. It runs on a sensor network organized in clusters. The choice of a cluster architecture is motivated by its efficiency in collaboration, data processing and transmission. The deployed network consists in powerful nodes playing the role of cluster heads and Camera Sensors as cluster members. Insofar as the mobility of the sensors is not required for the targeted applications, a static association algorithm is proposed to build the clusters. In the proposed tracking algorithm both intra and inter-cluster collaboration are possible.

In the case of a communicating target, the tracking algorithm starts when a Camera Sensor receives a periodic beacon from the mobile target. The information collected on this target is then sent to the Cluster Head, which selects a set of three close sensors in order to run the localization process with a Trilateration method [2]. The selection of these sensors is achieved using a probabilistic method. In addition to the clustering and the tracking algorithms, a deployment strategy for both cluster heads and members is proposed. It improves the tracking performances and ensure network connectivity.

To our best knowledge, our work is the first to integrate a collaborative target in the design of a tracking algorithm in the context of WMSN. Such target has only be considered in classical WSN [3] [4] or cellular networks [5].

The rest of this paper is organized as follows. In section II the pros and cons of the existing works are discussed. In section III, we detail the proposed tracking algorithm. Performances evaluation are discussed in section IV. Finally,

we conclude the paper in section V.

II. RELATED WORK

Previous research initiatives about target tracking in Wireless Sensor Networks can be classified into two main classes: proactive and reactive.

A. Proactive

Proactive tracking algorithms aim to estimate the upcoming target position based on the current one. To achieve this task predictive mechanisms are used.

In [6] the authors propose PTA: a Predictive Tracking algorithm in WMSN. It is a complete tracking algorithm that implements a five-step process: wake up, detection, localization, prediction, and next sensor selection. The main step being the prediction, it uses Kalman Filter as a predictive strategy. Kalman Filter is an efficient mathematical tool which uses current target coordinates to predict future ones.

An other Predictive Tracking strategy using Sequential Pattern is proposed in [7]. PTSP is based on two main steps: sequential pattern generation and object tracking. In the first step the prediction strategy is built based on object movement data collection. In the second step, the tracking starts. The concerned sensors are activated to continually keep tracking and then, missing targets are found. In [8] an exponential distributed predictive tracking protocol is proposed. With a lower computation complexity, it can estimate the target position. Then, using an Optimal Searching Radius (OSR), it computes the target searching radius.

The randomness of target behavior and trajectory can be difficult to anticipate. Consequently, proactive tracking algorithms cannot ensure the reliability of the tracking. An error in the estimation may causes a target missing.

B. Reactive

In reactive tracking algorithms the tracking is performed in reactively manner at each step of target movement inside the area of interest. Most of existing works in this category use a cluster-based network architecture. The clusters can be formed statically at the network deployment phase or dynamically during the tracking process. Two types of targets are considered in the following described solutions: non-communicating and communicating ones. In [9] a Bayesian estimation via Quantized Variational Filtering is used to select dynamically the cluster head at each step of tracking process. Once the cluster head is chosen, it selects a group of candidate nodes using a Multi-Objective Genetic Algorithm. This group is in charge of estimating the non-communicating target position. In [10], the authors propose a collaborative algorithm to address the node selection issue in WMSN. Considering a non-communicating target, the algorithm begins when the mobile object is detected by a Camera Sensor. This sensor becomes a cluster head and broadcasts its own location information to the nodes within its transmission range. Each of them computes the probability of detecting the object. If this probability

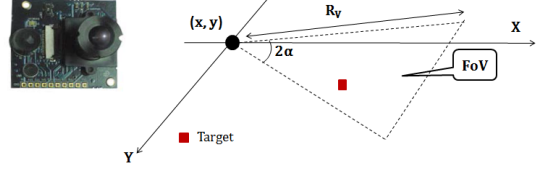


Fig. 1. Imote2 and Camera Sensor's Field of View

reaches a defined threshold, the node activates its camera to handle the localization phase.

In [3] and [4], the authors study the problem of tracking signal-emitting (communicating) mobile targets. In [3] the tracking is performed using mobile sensors. A sensor controller acquires the target signal's Time of Arrival (TOA) collected from the sensor network. Then, it estimates target location before directing the mobile sensor's path to follow and track the target. In [4] the authors study the tracking in its basic form. In other words, the output of each sensor is a simple "detected" or "not detected" information represented as 1 or 0 bit. Once the target is detected by a sensor, this sensor identifies the portion of arc of its circular sensing range that the target is crossing and the location of this target is fixed at the middle point of arc.

As in the lasts aforementioned works, we also consider a communicating mobile target in our proposed tracking algorithm.

III. A COLLABORATIVE TRACKING ALGORITHM FOR COMMUNICATING TARGET IN WMSN

In this section, we detail our contributions. After specifying the context, we present the deployment strategy for both cluster components. Then, the cluster association mechanism is described. Finally, the tracking algorithm with its four steps (detection, sensor selection, localization and target view) is presented. We propose a probabilistic model for the node selection process which is the key step of our tracking algorithm.

A. Preliminaries and assumption

A hierarchical, decentralized and cluster network architecture is set up. This choice is motivated by the need to ensure network scalability, traffic diminution and energy efficiency. The cluster members are a set of homogeneous and stationary multimedia sensors equipped with cameras. Lets call them CS for Camera Sensors. Each CS has a sector and directional Field of View (FoV) with opening angle 2α , video sensing radius R_V and transmission range R_T . As an example of existing Camera Sensor: MEMSIC (formallyCrossBow) proposes Imote2 sensors [11] which embed a low-resolution (640x480) cameras. An Imote2 and its FoV are shown in Fig.1. The CSs being limited in term of energy capabilities, we choose to deploy more powerful sensors without multimedia module as Cluster Heads. In order to minimize the CSs activity, the CHs are in charge of data collection, aggregation and routing. All CSs transmit their collected informations to the CH which

processes and forwards them to the sink. We assume that the sink can be located anywhere inside the area of interest. So, a multi-hop transmission between CHs is needed. To optimize the performances of the cluster-based architecture, a deployment strategy is necessary for both multimedia sensors and CHs. This strategy is described in the next sub-section.

B. Deployment strategy

As a preliminary step, the deployment strategy aims to maximize network video coverage as well as network connectivity. To achieve these tasks, we propose an enhanced version of the Virtual Force Algorithm (VFA) [12]. VFA uses repulsive and attractive forces to determine the new location of sensors. We assume that priori informations about the monitoring area are available. Consequently, we choose a planned initial deployment instead of random one. In planned deployment, the sensors are placed regularly following the area topology.

The deployment algorithm starts with the deployment of Camera Sensors. Then, the CHs are placed. Let our WMSN consists of N Cameras Sensors with IDs $\{CS_1, CS_2, \dots, CS_n\}$. After a planned deployment, each CS_i computes its total Force \vec{F}_i . \vec{F}_i represents the total repulsive and attractive forces applied on CS_i .

In the initial work, only the forces applied by the other CSs and obstacles are considered. In this work, we introduce the critical sub-areas concept and consider the forces applied by them. Let us explain the critical sub-area concept: in realistic environments, some areas are more important to monitor than others. For example in a building, entrances/exits and corridors have higher importance than individual offices. The critical sub-areas are weighted according to their importance. In this work, we do not study how to weight them; this is designed by an expert such as an architect. The weight attributed to each critical sub-area is considered in the deployment process. Considering this environment, \vec{F}_i is calculated as below:

$$\vec{F}_i = \sum_{\substack{j=1 \\ j \neq i}}^N \vec{F}_{ij} + \vec{F}_{obs} + W_{gt} \vec{F}_{csa} \quad (1)$$

where \vec{F}_{ij} represents the force applied between CS_i and CS_j . \vec{F}_{obs} is the total repulsive forces applied on CS_i by the surrounding obstacles. Finally, \vec{F}_{csa} is the total attractive or repulsive forces assigned on CS_i by the critical sub-areas. W_{gt} is the weight assign to each one. We specify then how to calculate \vec{F}_{ij} :

$$\vec{F}_{ij} = \begin{cases} (W_A(d_{ij} - d_{th}, \alpha_{ij})), & \text{if } d_{ij} > d_{th} \\ 0, & \text{if } d_{ij} = d_{th} \\ (W_R \frac{1}{d_{ij}}, \alpha_{ij} + \pi) & \text{if } \text{Otherwise} \end{cases} \quad (2)$$

W_A and W_R are respectively the measure of attractive and repulsive forces. α_{ij} represents the direction of \vec{F}_i . d_{ij} is the Euclidean distance between the gravity centers of CS_i and CS_j while d_{th} is the threshold distance which controls how close CSs get to each other. Its value is determined based on the sensing range R_V . With a similar reasoning, \vec{F}_{obs} and \vec{F}_{csa}

are calculated based on the distance between the sensor CS_i and the center of obstacles or critical sub-area

With the same working principal, we used W-VFA (Weighted VFA) to deploy the CHs. While the main target behind deploying CSs is to optimize the tracking, the main objective in CHs deployment is to ensure network connectivity. The CHs have omni-directional transmission range. Thus, \vec{F}_i is applied on the gravity center of the circular transmission range. Using the resulting deployment informations, each CH is aware of its final position and each CS is aware of its final position and camera orientation.

C. Association algorithm

After node deployment, the cluster-based association algorithm starts with a classical request/confirmation exchange. Each CH broadcasts a *Joint-request* containing its ID and its location. If a CS receives only one *Joint-request*, it sends back a *Joint-confirmation* to the concerned CH. A *Joint-confirmation* allows to associate definitively a CS to a CH, it contains CS's ID, location and final orientation. CHs location as well as CSs location and final orientation are obtained as output of the deployment phase.

A particular case may occur if a CS receives more than one *Joint-request* from different CHs. For more efficiency, the CH with the best LQI (Link Quality Indicator) is chosen. We choose a static clustering strategy instead of a dynamic one to ensure the balance between all the network components. Indeed, in dynamic clustering algorithms like LEACH [13] the clusters are formed at each round. Therefore, some sensors maybe exhaust their available energy too quickly due to being chosen as CH frequently. Furthermore, in the context of a static network, a dynamic clustering strategy is useless and has a non-negligible continuous network reconfiguration energy cost.

D. Tracking algorithm

The tracking process is divided in four steps: detection, sensor selection, localization and target view.

1) *Detection*: In most tracking applications, the targets apparition occurrence is asynchronous. Thus, putting all the sensors in active mode is unnecessary and too costly. In our work, after network deployment, all the Camera Sensors are in hibernation mode. A sensor is considered in hibernation mode (or deep sleep) when its sensing channel is inactive. The communication channel is kept in standby mode for communication and collaboration purposes. We assume that a single communicating target cross the area of interest by tacking a random trajectory. A communicating target is fitted with a communication component that allows it to collaborate with the network infrastructure. Consequently, it can be located anytime inside the deployment area. This target broadcasts periodically a *Beacon*. The tracking algorithm starts when a CS receives this last one. It measures the distance d_i between itself and the target using one of the existing Received-Signal-Strength (RSS) measurements techniques [14] [15]. Then, it informs its CH by sending a *Target-detected* message. This

message contains the sender ID, the value of d_i and the time of detection.

2) *Sensor selection*: Based on the information received from the CS via *Target-detected* message, the CH is in charge to elect two other sensors in its own cluster to localize the target using a Trilateration mechanism. The problem is how to select those sensors. Our solution is to use a probabilistic method at the CH.

Let us recall that the main objective of the proposed tracking algorithm is to handle the trade-off between the energy consumption and the tracking accuracy. The CS selection should be performed to meet this objective: for each CS, the CH computes the capability C_i of the tracking operation using equation (3).

$$C_i = \beta P_i + (1 - \beta) T_i \quad (3)$$

where P_i and T_i represent respectively the remaining power and the tracking accuracy of the i^{th} CS. β is the balancing parameter. Depending on its value, the priority is given to one of the evaluation metrics. The value of P_i is included in the interval $[0,1]$ where 1 represents 100% of remaining energy. The information of the remaining energy is sent by the cluster members when a change on this value occurs. The value of T_i is also included in the interval $[0,1]$ and is obtained as detailed below:

$$T_i = 1 - (D_i/R_T) \quad (4)$$

D_i is the distance between the i^{th} CS and the target. The temporarily target location is calculated by the CH based on the distance d_i sent by the first CS that detects the target as detailed above. R_T as mentioned in III.A is the transmission range and thus, the maximal distance beyond which the CS cannot detect or localize the target. If D_i is higher or equal to R_T , the i^{th} CS cannot localize the target. Consequently, T_i is equal to zero.

Finally, the probability Pr that a node would be selected to perform target localization is obtained as shown in equation (5):

$$Pr_i = 1 - (N_i/N_c) \quad (5)$$

where N_c denotes the number of cluster members within the cluster. N_i is the number of cluster members within the same cluster having a higher capability C_i than the i^{th} CS.

The cluster members with the highest probability are selected by the CH to participate to the localization process. Once the CH selects the best CSs to support it in target localization process, it informs them by sending a *Localization* request. Each of the selected one measures the distance d_i (if it is not already available) between itself and the target by requesting a *beacon* from it. Then, it replies to the CH with *Target-located* message which contains this distance. In the case where one of the selected CSs cannot compute the distance d_i . The CH repeats the process until it selects three sensors able to compute their respective d_i .

In the tracking process a particular case may occur. A target can be simultaneously detected by more than one cluster. If it is so, it results data redundancy and a loss of resources. As

Algorithm 1 Algorithm running at each Camera Sensor

```

1: switch (case)
2: case 1 Target detected:
3:   Measure the distance  $d_i$ ;
4:   Send Target-detected message to CH;
   Break
5: case 2 Localization request by CH :
6:   Perform target localization;
7:   Send Target-location message;
8: case 3 Camera-activation request by CH :
9:   Capture target's frames;
10:  Send multimedia-data message;
   Break
11: end switch

```

Algorithm 2 Algorithm running at each Cluster Head

```

1: switch (case)
2: case 1 Receive Target-detected message:
3:   Select the 3 best located Camera Sensors;
4:   Send Localization request to those Camera Sensors;
5:   Select the best oriented Camera;
6:   Send Camera-activation to the selected Camera Sensor;
7:   Send the multimedia data to the sink during the network
   inactive periods;
   Break
8: case 2 Receive Target-location message :
9:   Compute target position using Trilateration;
10:  Send the result to the sink during the network inactive
   periods;
   Break
11: end switch

```

a solution, we propose an inter-cluster collaboration. Indeed, while in the classical cases the CH collaborates only with its cluster members, in this case the CH can exchange information with other neighbors CHs to improve the algorithm's performances. The approach is as follows: if a CH receives a *Target-detected* message sent from a border CS. It informs the closest CH to the border CS by sending a *Target-detected-bis* request. Let us specify that a CS is considered as a border one by its CH if the distance $d(CH, CS)$ between them satisfies the following inequality:

$$d(CH, CS) > R_{CH_T} - R_V \quad (6)$$

R_{CH_T} is the CH transmission range. This inequality ensures that the CSs which have all their field of view inside the CH transmission range are not considered as border ones. The *Target-detected-bis* request contains the location of the border CS, the measured distance d_i and the time of target detection. When the closest CH receives this request, it firstly checks if there is a CS in its own cluster which detects a target at the same time. If so, it proceeds to node selection as usually and sends back a *sensor-selection* message containing the location of the three selected CSs in its cluster to perform Trilateration.

When the initial CH receives these informations it selects the three best located sensors among its cluster members and the three CSs of the other cluster. If more than one CS is located in its cluster, it handles the rest of the tracking process. Otherwise, it sends to the other CH a *localization-bis* request. Thus, this last one is in charge of the tracking.

3) *Localization*: To achieve target localization, the CH uses a Trilateration algorithm [2]. The basic idea of the used algorithm is that every point inside a triangle can be calculated by knowing the coordinates of the triangles vertices's. The procedure is as follows: We assume that the coordinates of the target T are (x_T, y_T) . The coordinates of the three selected CSs are (x_1, y_1) , (x_2, y_2) , (x_3, y_3) and finally, the distances between them and T are d_1, d_2, d_3 respectively. The position of T can be calculated by solving an array of non linear equations [16]:

$$(x_T - x_i)^2 + (y_T - y_i)^2 = d_i^2 \text{ for } i = 1, 2, 3 \quad (7)$$

4) *Target view*: After target localization, the CH selects the best oriented Camera to activate using Target in Sector test [17]. This test aims to check if the target is really in CS's field of view. This Camera belongs to one of the three selected CSs involved in the localization process. Each CH is aware of the orientation of its cluster members via the *Joint-confirmation* message exchanged with the CSs during the clustering phase. The camera is used to observe, identify and classify the target. These multimedia data can be used to provide personalized services and applications to a specific class of targets. Moreover, the data can be very useful to achieve multi-target tracking. The CH informs this selected CS by sending a *Camera-activation* request. This CS replies by a *multimedia-data* message containing the target frames captured. This optional step is executed only if the CS has enough residual energy to accomplish it.

Finally, all the data and results are sent to the sink by the CH during the inactive network periods. Algorithms 1 and 2 describe the pseudo code of the proposed solution running respectively at the Camera Sensors and the Cluster Heads.

IV. PERFORMANCE EVALUATION

In this section, we discuss simulation settings, performances metrics and results. To evaluate the performances of the proposed algorithm, we have used NS-2 simulator [18]. Table I summarizes the simulation parameters used. The values of R_T , R_V and α are selected according to real Imote2 features [11]. Five random target trajectory as well as three evaluation metrics are also used to evaluate our work: tracking accuracy, energy consumption and the volume of message exchanged. The value of the parameter β used in equation (3) is set to 0.5, it allows to balance the impact of both energy consumption and tracking accuracy.

Although there are many papers considering communicating targets, our work is the first to integrate a collaborative target in the design of a tracking algorithm in the context of WMSN. Indeed, such target has only be considered in classical WSN [3] [4] or cellular networks [5] where the constraints related

to the kind of network are different. Thus, we cannot compare our solution to those mentioned before.

TABLE I
PARAMETER VALUES OF THE SIMULATION

Phy and Mac Layers	IEEE 802.15.4
Access Mode	CSMA/CA non-beaconed
Area size	200m x 200m
Target speed	1.38 m/s (pedestrian)
Number of Camera Sensors	20, 30, 40, 50
Number of Cluster Heads	5, 10, 15, 20
Simulation Time	200 s
Transmission range (R_T)	30 m
Depth of view (R_V) of CS	20 m
Angle of view of CS (2α)	$\pi/3$
CH transmission range (R_{CHT})	60 m
Size of messages	100 bytes
Balancing parameter β	0.5

TABLE II
ENERGY MODEL PARAMETERS (IMOTE2)[11]

Initial node energy	3 AAA
Active power	0.279 Joule/Second
Idle power (Radio on Camera off)	0.226 Joule/Second
Sleep power (Radio off Camera off)	0.015 Joule/Second
rx/tx Power (Frequency 104 MHz)	0.078 Joule/Second
Camera Power	0.044 Joule/Second

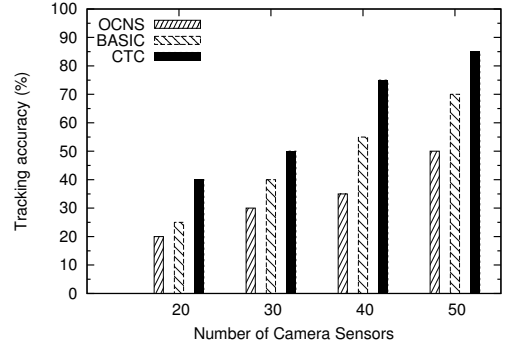


Fig. 2. Tracking accuracy vs. number of Camera Sensors

Instead, we compare the results obtained by our algorithm with two other ones where only camera sensors are deployed: 1) BASIC, where all the nodes are always in active state, the localization is performed using an image processing solution [19] and 2) OCNS an other reactive and cluster-based solution described in [10], more details are given in section II. For the rest of the paper, we refer to our proposed solution as CTC for Cluster-based Tracking algorithm for Communicating target.

A. Tracking Accuracy

To calculate tracking accuracy, we consider the number of location points obtained by the algorithm along a defined target trajectory. Based on the size of the area of interest, target velocity, the number of deployed sensors and their sensing range, the best tracking accuracy (100%) is reached when one location point is reported every five meters.

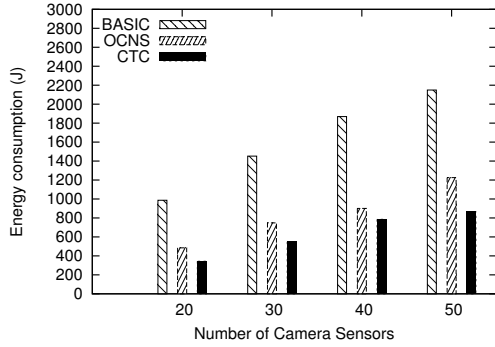


Fig. 3. Energy Consumption vs. number of Camera Sensors

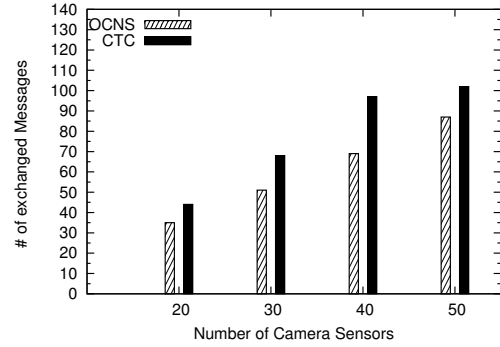


Fig. 4. Exchanged messages vs. number of Camera Sensors

Fig.2 illustrates the tracking accuracy vs. the number of Camera Sensors. Due to the cameras that are always active, BASIC has better performances than OCNS. However, CTC algorithm obtains the best performances. It increases the tracking accuracy by up to 40% compared to the solutions where a non-communicating target is considered. Indeed, the collaborative target emits signals to facilitate both the detection and the localization steps. Moreover, unlike in BASIC and OCNS where the Camera Sensors are scattered randomly in the area of interest, the sensors (CSs and CHs) are deployed following W-VFA strategy described in section III.B. We conclude that the deployment strategy has a positive impact on tracking performances. This impact is more important than the number of active CSs. OCNS has the worst results because of the wake up and relaying strategy. A target can enter the monitoring area without being detected instantaneously.

B. Energy Consumption

We use the energy consumption model of Imote2 sensor nodes [11] as reference data for our simulations. The parameters are indicated in Table II. To evaluate energy consumption, we compute the energy cost of camera activation, active period duration, localization and communication cost during the tracking process. Fig.3 depicts the mean energy consumption of the network in Joule (J) vs. the number of Camera Sensors deployed. Clearly, BASIC is an inconsiderable solution. Because of the permanent active state of the CS, it consumes up to 2148.63 J. CTC has better results compared to OCNS. We explain them by the use of more powerful Cluster Heads with unlimited energy constraint which handles the high cost processing and the collaboration tasks while the CSs are only in charge of the low cost tasks.

C. Exchanged Messages

The number of exchanged messages represents the communication overhead. We calculate it by considering the volume of collaborative messages exchanged during the tracking process. Due to the low node density used in the simulation scenarios, no collision or packet loss is recorded. Fig.4 shows the average number of exchanged messages vs. the number of Camera Sensors. BASIC scheme is not illustrated in the figure because there is no collaboration between the sensors.

We observe that the communication cost of OCNS is less than CTC. Indeed, while in OCNS only inter-cluster communication is possible, in CTC algorithm both inter and intra-cluster collaboration are feasible. Moreover, its better performances in tracking accuracy has a non negligible communication cost. However, even if the amount of exchanged messages for CTC algorithm is higher than the compared solutions, it always consumes less energy.

V. CONCLUSION

Target tracking is a very useful application. However, in the context of WMSN, it can be a challenging task due to the resources constraint. In this paper, we propose a cluster-based tracking algorithm which handles the trade-off between the energy conservation and the tracking performances. The tracking is achieved based on the collaboration between the different component of the network. High cost tasks are handled by powerful cluster heads while low-cost tasks are handled by constraint cluster members. Simulation studies prove the efficiency of the proposed algorithm. Indeed, it allows to increase the tracking accuracy by up to 40% and save up to 1284.83J energy compared to other algorithms. Open issues for future research include tracking delay analysis depending on target speed.

REFERENCES

- [1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer networks*, vol. 51, no. 4, pp. 921–960, 2007.
- [2] Y. Wang, X. Wang, D. Wang, and D. P. Agrawal, "Range-free localization using expected hop progress in wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 20, no. 10, pp. 1540–1552, 2009.
- [3] E. Xu, Z. Ding, and S. Dasgupta, "Target tracking and mobile sensor navigation in wireless sensor networks," *IEEE Transactions on mobile computing*, vol. 12, no. 1, pp. 177 – 186, 2013.
- [4] Z. Wang, E. Bulut, and B. K. Szymanski, "Distributed energy-efficient target tracking with binary sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 4, p. 32, 2010.
- [5] M. M. Olama, S. M. Djouadi, and C. S. Pendley, "Position and velocity tracking in mobile cellular networks using the particle filter," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Las Vegas, USA, 2006.
- [6] I. Boulanouar, S. Lohier, A. Rachedi, and G. Roussel, "PTA: A predictive tracking algorithm in wireless multimedia sensor networks," in *International Conference on Global Information Infrastructure and Networking Symposium (GIIS)*, Trento, Italy, To appear.

- [7] S. Samarah, M. Al-Hajri, and A. Boukerche, "A predictive energy-efficient technique to support object-tracking sensor networks," *IEEE Trans on Vehicular Technology*, vol. 60, no. 2, pp. 656–663, 2011.
- [8] L. Xue, Z. Liu, and X. Guan, "Prediction-based protocol for mobile target tracking in wireless sensor networks," *Systems Engineering and Electronics, Journal of*, vol. 22, no. 2, pp. 347–352, 2011.
- [9] M. Mansouri, L. Khoukhi, H. Nounou, and M. Nounou, "Genetic algorithm optimization for quantized target tracking in wireless sensor networks," in *Global Telecommunications Conference (GLOBECOM)*, Houston, USA, 2011.
- [10] L. Liu, X. Zhang, and H. Ma, "Optimal node selection for target localization in wireless camera sensor networks," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 7, pp. 3562–3576, 2010.
- [11] Imote2. [Online]. Available: <http://www.memsic.com>
- [12] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *INFOCOM*, San Francisco, USA, 2003.
- [13] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the Hawaii International Conference on System Sciences*, Hawaii, USA, 2000.
- [14] G. Mao, B. Fidan, and B. Anderson, "Wireless sensor network localization techniques," *Computer networks*, vol. 51, no. 10, pp. 2529–2553, 2007.
- [15] S. Fernández, D. Gualda, J. C. García, J. J. García, J. Ureña, and R. Gutiérrez, "Indoor location system based on zigbee devices and metric description graphs," in *IEEE International Symposium on Intelligent Signal Processing (WISP)*, Floriana, Malta, 2011.
- [16] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, "GPS theory and practice," 1997.
- [17] Y. E. Osais, M. St-Hilaire, and R. Y. Fei, "Directional sensor placement with optimal sensing range, field of view and orientation," *Mobile Networks and Applications*, vol. 15, no. 2, pp. 216–225, 2010.
- [18] The NS-2 website. [Online]. Available: <http://www.isi.edu/nsnam/ns/>
- [19] H. Oztarak, K. Akkaya, and A. Yazici, "Efficient localization and tracking of multiple objects in wireless multimedia sensor networks," *Ad Hoc and Wireless Sensor Networks*, vol. 19, 2013.